IJECBE (2025), **3**, **4**, 786–801
Received (9 December 2025) / Revised (22 December 2025)
Accepted (22 December 2025) / Published (30 December 2025)
https://doi.org/10.62146/ijecbe.v3i4.206
https://ijecbe.ui.ac.id
ISSN 3026-5258

International Journal of Electrical, Computer and Biomedical Engineering

RESEARCH ARTICLE

Securing GTP Protocol on Cellular Networks using Anomaly Based Intrusion Detection System

Muhammad Fikriansyah and Alfan Presekal*

Department of Electrical Engineering, Faculty of Engineering, Universitas Indonesia, Depok, Indonesia
*Corresponding author. Email: presekal@.ui.ac.id

Abstract

Cellular networks are growing rapidly and have become critical infrastructure. Along with this growth, threat actors increasingly target both network operators and subscribers for financial gain, espionage, and data theft. To reduce these risks, reliable detection mechanisms are needed to identify malicious traffic within cellular networks. This study proposes an anomaly based intrusion detection system using a convolutional neural network (CNN) to secure the General Packet Radio Service Tunnelling Protocol (GTP). A GTP specific dataset containing normal and malicious traffic was constructed from a controlled simulation environment. The extracted traffic features were used to train and evaluate the CNN model for binary attack detection. The proposed CNN model achieved 99.07% accuracy, 98.59% precision, 99.43% recall, and an F1-score of 99.00%.

Keywords: Intrusion Detection System, Convolution Neural Network, General Packet Radio Service Transport Protocol

1. Introduction

Cellular networks have evolved into critical infrastructure, carrying personal communications, personal identity, financial transactions, industrial IoT traffic, and public services. At the same time, cellular networks face increasing threats from signalling abuse and user plane hijacking. Surveys on mobile and 5G security consistently show that the attack surface has expanded with softwarisation, virtualisation and service based architectures, making core networks and tunnelling protocols attractive targets for sophisticated adversaries [1] [2] [3]. On the other hand, cellular networks still contain protocol level vulnerabilities. For instance, the GPRS Tunnelling Protocol

(GTP) does not provide authentication between network elements. The protocol also has threats such as subscriber denial of service, subscriber tracking, billing fraud or network distributed denial of services. Intrusion Detection Systems (IDS) are largely used in modern 4G/5G deployments. IDS designs are typically grouped into signature based and anomaly based. Signature based detection matches traffic against known attack patterns stored in signature databases, whereas anomaly based systems learn normal behaviour and flag deviations from it [2],[4]. Signature based IDSs are efficient and precise for known threats, but they often struggle to detect zero day attacks. Anomaly based IDS can catch novel attacks but have risks to false positives, especially in dynamic mobile environments. Recent 5G security surveys argue that no single detection paradigm is sufficient—hybrid, multi layer designs are needed to cope with heterogeneous traffic and strict latency constraints [2], [5]. To improve robustness, many recent works combine these traditional ideas with artificial intelligence (AI) and machine learning (ML). ML based IDS models using deep neural networks, recurrent or convolutional architectures, and more recently specialised spiking or hybrid models—have been demonstrated for various 5G scenarios, including SDN based 5G cores and device to device (D2D) communications in IoT [5],[6],[7]. These systems typically learn from labelled intrusion detection datasets and can capture complex, non linear decision boundaries, improving detection rates for subtle or previously unseen attacks. Existing research dataset like CIC-IDS2017 and CSE-CIC-IDS2018 contain rich traffic and multiple attack types [8],[9]. Another 5G datasets that is 5G-NIDD provide labelled traffic from an operational 5G test network and are used to evaluate ML based IDS for non IP data delivery [10]. Meanwhile, 5GCIDS datasets focuses on PFCP traffic in the 5G core and offers an AI based IDS together with a protocol specific dataset [11]. All of the dataset above, are lack of the GPRS Tunnelling Protocol (GTP) traffic in LTE/5G networks. This study addresses these gaps by focusing specifically on GTP security in mobile networks. The objectives are:

1. To create a new GTP specific dataset that includes labelled benign and malicious GTP-U traffic including flood, invalid TEID, spoofing and malformed attacks.
2. To develop CNN based models that can classify the GTP normal traffic and malicious traffic.
3. To evaluate the models under realistic traffic scenarios and varying attack intensities, using standard metrics such as accuracy, precision, recall and false positive rate.

2. Cellular Networks

2.1 Architecture

Mobile networks have traditionally been treated as a closed, "walled garden" environment, often perceived as isolated and difficult to attack directly from the Internet [3]. In practice, however, the core network can be reached both from the public Internet and via roaming partners. Figure 1 presents the core architectures of LTE and 5G NR Standalone (SA). Abbreviations used:
MME - Mobility Management Entity
SGW - Serving Gateway
PGW - Packet Data Network Gateway

PCRF - Policy and Charging Rules Function
AMF - Access and Mobility Management Function
SMF - Session Management Function
UDM - Unified Data Management
AUSF - Authentication Server Function
GRX/IPX - GPRS Roaming Exchange / IP Exchange
PCF - Policy Control Function
UPF - User Plane Function
SIEM - Security Information and Event Management
GTP-C - GPRS Tunnelling Protocol, Control Plane
GTP-U - GPRS Tunnelling Protocol, User Plane

Figure 1 highlights the structural differences between the 4G LTE and 5G SA core. In LTE, GTP-C carries control plane signalling among the MME, SGW, and PGW, whereas GTP-U transports user-plane traffic. In the 5G SA service based architecture, the AMF and SMF assume the roles of the MME, and user plane connectivity over the N3/N4 interfaces still relies on GTP-U. The Intrusion Detection System (IDS) proposed in this work is positioned to observe GTP flows at key points in the network, namely: between the SGW and PGW in LTE (S5/S8 interface), between the SMF and UPF in 5G SA (N4 interface), and at roaming interconnection links (S8 for LTE and N9 for 5G).

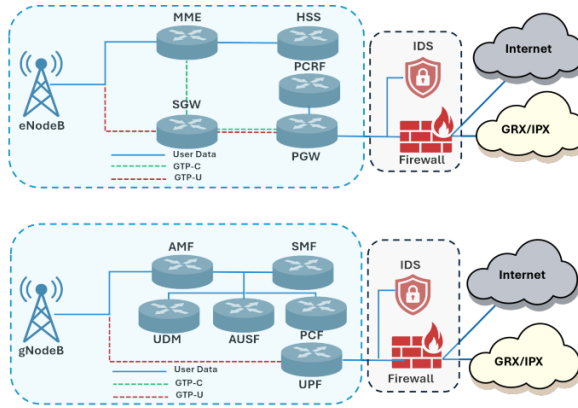

Figure 1. Cellular Network Architecture

2.2 Threats and Exploitation

In August 2015 a security researcher published ways to attack mobile network operator (MNO) via the GRX/IPX network and the internet using GTP-C messages [12]. This was the trigger for a systematic risk analysis within some MNOs to determine the attack potential, the risks that go with the identified attacks and the gain for the attacker. As a result, countermeasures that are effective for protecting a mobile network and its subscribers were derived. A variety of attacks have already been

successfully performed on production network equipment of MNOs. Exposure of network elements to these attacks allows attackers to exploit the mobile network by reconfiguring network elements for their purposes, by using network elements that are controlled by attackers to jump deeper inside the network to other network elements, by decrypting subscriber's packet data sessions, and by hijacking existing internet sessions of subscribers.

2.2.1 GTP Threats

It describes several types of threats, including information disclosure, denial of service against subscribers, and subscriber tracking. The related attack techniques are as follows: [13]

1. Network Denial of Service Against All Subscribers.
   In this scenario, the attacker floods the PGW via the GRX with a large number of Create Session Request messages. Each request is crafted to look like a valid attempt to establish a data session. For every such request that appears to originate from a different user, the PGW allocates a fresh IP address from its DHCP pool. The attacker repeatedly sends these session requests in large volumes over a short period. Because the PGW does not enforce sufficient validation or rate limiting, it continues assigning IP addresses to every incoming request. Eventually, the DHCP address pool at the PGW is exhausted. Once this happens, legitimate subscribers who try to start a data session can no longer obtain an IP address and instead receive a "No resource available" error.

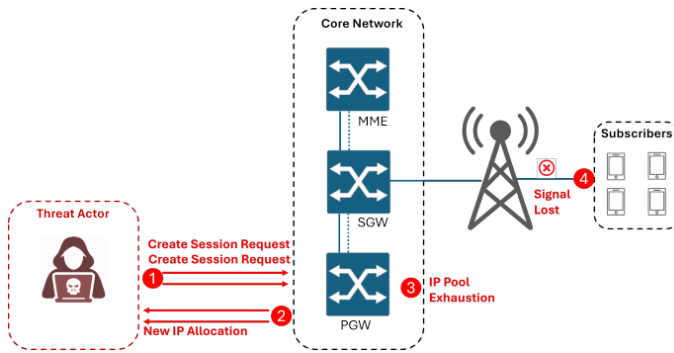

Figure 2. Network DoS

2. Subscribers Denial of Service.
   Attackers can perform a simple subscriber DoS attack using Delete Session Request, Delete Bearer Request to PGW, or Delete PDP Context Request to GGSN via external S8 and Gp interface using TEID. It could be a massive subscriber DoS in case multiple messages are sent with different TEIDs. [14]
3. Billing Fraud This attack is conceptually simple but particularly damaging. The attacker sends a Create Session Request or Create PDP Context Request that includes a spoofed or stolen IMSI (International Mobile Subscriber Identity).

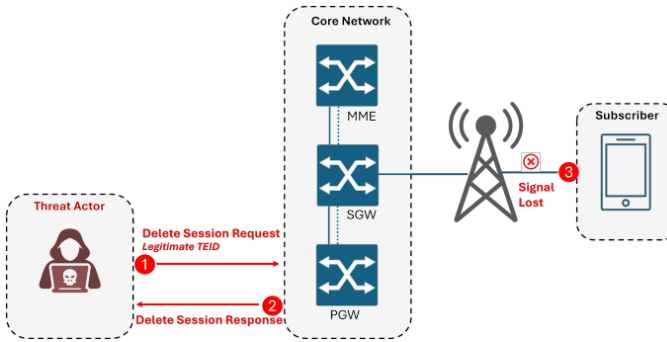

Figure 3. Subscriber DoS

When the network processes this request, it binds the resulting data session to the victim's IMSI instead of the attacker's identity. Consequently, all data usage and services consumed during this fraudulent session are charged to the victim's account. The legitimate subscriber is billed for traffic they never generated, while the attacker enjoys unauthorized access to network services without bearing any cost

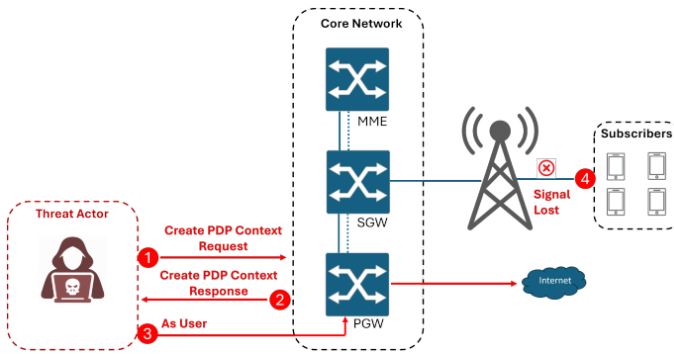

Figure 4. Billing Fraud

4. Subscriber Location Tracking This method relies on continuously reading User Location Information (ULI) carried in MS Info Change Notification Request messages exchanged between the VPLMN (Visited Public Land Mobile Network) and the HPLMN (Home Public Land Mobile Network). Under normal conditions, as a subscriber moves, their location updates are propagated through GTP signalling between the visited and home networks. An attacker who manages to intercept this signalling can quietly monitor these messages and obtain near real time updates on the subscriber's location whenever it changes.

3. Methodology

This section presents the methodology of the proposed anomaly based intrusion detection system for LTE/NR cellular networks. The methodology consists of simulation setup, traffic generation, dataset composition, system architecture overview, and anomaly based detection. A convolutional neural network (CNN) was used as the main machine learning model.

3.1 Simulation Setup

The simulation is run through virtual machines. First virtual machine runs UERANSIM, which emulates both the User Equipment and the gNodeB. Second virtual machine runs the Open5GS core components, including the Access and Mobility Management Function, Session Management Function and User Plane Function, as well as MongoDB for storing subscriber and session data. As the simulated mobile device establishes connectivity via the simulated base station, control signaling messages are exchanged between the AMF and SMF over the control plane, whereas actual data transmission is processed by the UPF utilizing the GTP-U protocol. Throughout these network interactions, Wireshark captures all packet transmissions, which are subsequently stored in PCAP file format. The simulation also includes attack scenarios and unusual traffic patterns to make the data more varied. This method makes both real and fake GTP protocol traffic, giving researchers useful datasets to use when creating and testing intrusion detection systems and algorithms for finding unusual behavior that are meant to secure mobile core network infrastructure.

3.2 Traffic Generation

The traffic generation process in this study consisted of two categories, namely normal traffic and attack traffic. First, Normal traffic represents typical user traffic found on a network. The normal traffic scenarios used are as follows:

1. Open Web Page by HTTP GET
2. Download 10MB files by HTTP
3. Sending ICMP Packets to 8.8.8.8

Second, Attack traffic is created to simulate various attack scenarios on a network. In this study, four attack scenarios were used, as follows:

1. Flood Attack, namely an attack carried out by sending GTP-U packets repeatedly to overload the network resources or target node, thus potentially reducing service availability
2. Invalid TEID, namely an attack by sending GTP-U packets with invalid TEID values or not registered with the active tunnel on the destination node
3. Spoofing, namely an attack by sending packets as if from a legitimate source IP and tunnel, but in the request and response sections it appears to use different MAC addresses
4. Malformed, namely sending a GTP-U packet with an invalid or incomplete protocol structure, so that it cannot be interpreted normally by the receiving system.

The generated traffic was then organized into labelled classes for model development. To improve reproducibility, Table 1 summarizes the number of samples used in the training, validation, and testing stages for each traffic class.

3.3 Dataset Composition and Class Distribution

Table 1. Dataset composition and class distribution used in this study

Traffic	Class	Train	Test
Normal	18,562	3,978	3,978
Attack	16,614	3,560	3,560
Total	35,176	7,538	7,538

As shown in Table 1, the dataset consisted of 50,252 samples, including 26,518 normal samples and 23,734 attack samples. The data were divided into training, validation, and test sets using a stratified 70:15:15 split to preserve the class proportion across all subsets.

3.4 System Architecture Overview

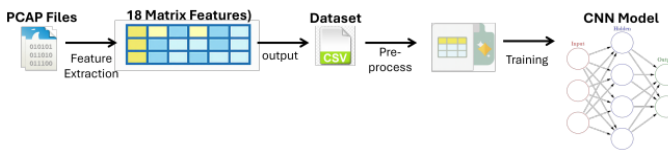

Figure 5. System Architecture Overview

This diagram shows an anomaly based Intrusion Detection System (IDS) that uses a Convolutional Neural Network (CNN) architecture to find unusual patterns in network data. The system starts with PCAP (Packet Capture) files that contains raw network traffic data in binary format. These raw packets are then put through a feature extraction method that changes the binary packet data into a matrix format with 18 features. These features include network traffic characteristics such as packet size, protocol types, flow duration, byte counts, flag patterns, and timing information, converting unstructured packet data into quantifiable metrics suitable for machine learning analysis.

The extracted characteristics are put into a structured matrix with rows for each network flow or connection and columns for the 18 various feature dimensions. This makes a standard representation that makes it possible to recognize patterns across different traffic samples. Then, this feature matrix is saved in CSV format, which makes a permanent dataset that can be used for training, data validation, augmentation, and reuse in different training sessions. Before training, the dataset undergoes preprocessing steps including normalization to scale features to consistent ranges, handling missing values, balancing class distributions between normal and anomalous traffic, and potentially splitting into training, validation, and test sets.

The preprocessed data is then used to train a CNN architecture consisting of an input layer that receives the 18 feature vectors, multiple hidden layers with convolutional and pooling operations that automatically learn hierarchical feature representations and detect spatial patterns in network behavior, and an output layer that produces classification results distinguishing between normal and anomalous traffic or identifying specific attack types. This system operates as an anomaly based IDS because it learns patterns of normal network behavior during training and identifies deviations from these learned patterns as potential intrusions.

3.5 Anomaly Based Detection

This approach analyzes network traffic through spatial feature patterns utilizing Convolutional Neural Networks (CNN), a methodology that has also been explored in previous research [15] [16] [17] [18].

1. FEATURE EXTRACTION
   Feature extraction was performed by converting each packet into 18 numerical features that represent its main protocol and traffic characteristics. These features were taken from the IP, UDP, GTP headers, packet payload and traffic statistics. In this way, each packet can be represented in a consistent form before being used as input to the CNN model.

Table 2. Feature categories used in the proposed method

Feature Category	Features	Count
Packet Parameters	(1) Packet size; (2) IP total length; (3) time-to-live (TTL); (4) IP protocol number; (5) fragmentation offset; (6) fragmentation indicator.	6
Transport Layer Parameters	(7) UDP source port; (8) UDP destination port; (9) UDP length.	3
GTP Protocol Parameters	(10) Tunnel Endpoint Identifier (TEID); (11) GTP message type; (12) sequence number.	3
Payload Characteristics	(13) Payload length; (14) payload entropy.	2
Statistical Features	(15) GTP presence indicator; (16) packet rate; (17) byte rate; (18) inter-arrival time.	4
Total		**18**

The selected features were chosen to capture both packet structure and traffic behavior. Header based features, such as packet size, TTL, port numbers, TEID, message type, and sequence number, describe the protocol structure of each packet. Payload based features, including payload length and entropy, help reflect differences in content characteristics. Statistical features, such as packet rate, byte rate, and inter-arrival time, were included to represent temporal behavior that may indicate flooding or other abnormal traffic patterns.
The feature selection process was carried out in a domain driven manner. We first identified candidate fields that were directly available from the packet capture and relevant to GTP traffic analysis. From these candidates, we retained features

that were meaningful for detecting the attacks considered in this study, could be extracted consistently from all samples, and were lightweight enough for practical use. On the other hand, fields such as raw IP addresses, MAC addresses, and other environment specific identifiers were excluded because they may introduce bias and reduce the generalizability of the model. Based on this process, the final 18 features were used as the input representation for the proposed CNN.

2. DATA PREPROCESSING

After feature extraction, we apply Z-score normalization so that each feature has zero mean and unit variance on the training dataset. For a given feature value x, the standardized value x^{norm} is computed using Eq. 1, where μ and σ denote the mean and standard deviation of that feature over all training samples. These normalization parameters are estimated once during training using verified benign traffic and then reused to normalize all incoming traffic during deployment. The mean σ_i of feature i over m training samples is defined in Eq. 2, where x_{ij} is the value of feature i in sample j, and $\sum_{j=1}^{m}$ denotes the summation over all samples. The corresponding standard deviation $\sum_i$ is computed as in Eq. 3, where $(x_{ij} - \sigma_i)^m$ is the squared deviation from the mean. Finally, the normalized feature matrix element is given in Eq. 4, where x_{ij}^{norm} denotes the standardized value of feature i in sample j.

$$x^{norm} = \frac{x - \mu}{\sigma} \tag{1}$$

$$\mu_i = \frac{1}{m} \sum_{j=1}^{m} x_{ij} \tag{2}$$

$$\sigma_i = \sqrt{\frac{1}{m} \sum_{j=1}^{m} (x_{ij} - \mu_i)^2} \tag{3}$$

$$X_{ij}^{norm} = \frac{x_{ij} - \mu_i}{\sigma_i} \tag{4}$$

3. CNN FOR SPATIAL ANALYSIS

Normal GTP traffic tends to follow stable and predictable patterns. For example, certain message types are typically associated with specific TEID ranges, payload sizes are constrained by protocol rules, and port combinations follow expected communication flows. The joint (spatial) distribution of these features forms characteristic signatures that differ between legitimate traffic and attack traffic. We use a Convolutional Neural Network (CNN) to capture these spatial anomalies. CNNs learn how features depend on and interact with each other by using convolutional filters. This is different from fully connected networks, which process each feature mostly on its own. These filters go throughout the feature space and automatically find patterns that differentiate normal traffic from strange activity. This makes CNNs a good choice for multi-dimensional feature vectors that come from network flows.

The CNN architecture in this study consists of two convolutional blocks followed by dense layers for binary classification. The first block uses 32 filters with kernel size 3, followed by batch normalization, max-pooling, and dropout with a rate of 0.5 for regularization. The second block uses 64 filters with the same kernel size and batch-normalization-max-pooling structure, but with a higher dropout rate of 0.6. After flattening, the extracted spatial features are passed through two fully connected layers with 64 and 32 units, each accompanied by dropout (0.6), before reaching a final sigmoid output layer that produces the binary classification score. The model is trained in a supervised manner on a dataset containing both benign and malicious traffic samples. Because the class distribution is slightly imbalanced, class weights are applied during training to reduce bias toward the majority class. Training uses binary cross-entropy loss with the Adam optimizer (learning rate 0.0005) and L2 regularization (0.01) to reduce overfitting. Through backpropagation over 100 epochs with early stopping based on validation loss, the network learns to map input feature patterns directly to benign/attack labels.
The input to the CNN is a feature vector that is reshaped into a one-dimensional sequence suitable for Conv1D operations. For n features extracted from each traffic flow, the vector is reshaped to $(n, 1)$, preserving the sequential ordering of features so that convolutional filters can capture local correlations between adjacent dimensions. The output layer produces a prediction score $P\epsilon\ [0, 1]$ by applying the sigmoid activation in Eq. 5, where z is the weighted sum from the final dense layer. The spatial anomaly score $E_{spatial}$ is then defined as in Eq. 6, where values close to 1 indicate malicious traffic and values close to 0 indicate benign traffic.

$$P = \sigma(z) = \frac{1}{1 + e^{-z}} \tag{5}$$

$$E_{spatial} = P \tag{6}$$

4. ANOMALY THRESHOLD DETERMINATION
After computing the spatial anomaly score for each traffic flow, a decision threshold is required to separate benign traffic from attacks. We use Receiver Operating Characteristic (ROC) curve analysis to figure out this best threshold. We test a number of threshold settings on a labeled dataset that has both good and bad traffic throughout the validation phase. We find the True Positive Rate (TPR) and False Positive Rate (FPR) for each possible threshold. We define the following for each threshold τ:

a) True Positive (TP): the number of attack flows that were correctly identified as attacks,
b) False Positive (FP): the number of benign flows that were incorrectly identified as attacks,
c) True Negative (TN): the number of benign flows that were correctly identified as benign;,
d) False Negative (FN): the number of attack flows that were incorrectly identified as benign.

Using these quantities, the True Positive Rate (TPR) and False Positive Rate (FPR) at threshold τ are computed as

$$TPR(\tau) = \frac{TP(\tau)}{TP(\tau) + FN(\tau)'} \tag{7}$$

$$FPR(\tau) = \frac{FP(\tau)}{FP(\tau) + TN(\tau)\cdot} \tag{8}$$

4. Result

This section describes the performance evaluation of proposed anomaly based intrusion detection system.

4.1 Precision, Accuracy, Recall, F1-Score, AUC

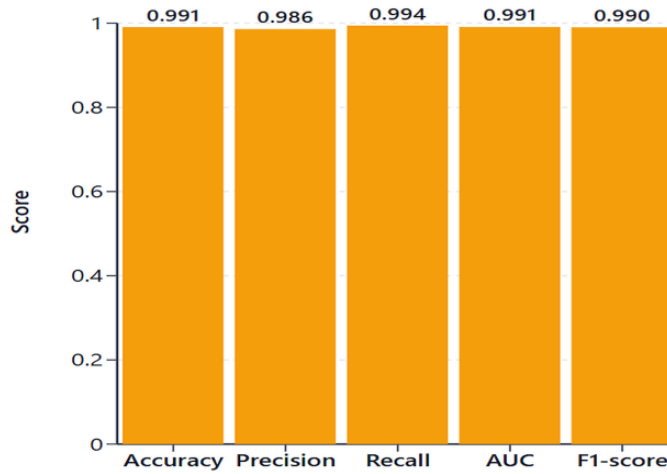

Figure 6. Precision, Accuracy, Recall, F1-Score, AUC

This bar chart shows five important performance measures that were looked at on the test dataset. The model accurately classifies around 19 out of every 20 occurrences overall, which is a 99.1% accuracy rate. With a precision of 98.6%, the model is right around 9 times out of 10 when it predicts the positive class. This means that there are some false positives, but they are not very common. The recall of 99.4is especially excellent because the model finds almost all real positive cases, missing only 1 or 2. The AUC score of 99.1% backs up the ROC curve analysis, showing that the model can tell the difference between things quite well. The F1-score of 99.0% is the harmonic mean of precision and recall, which shows that the two measures are well-balanced. The model seems to be more cautious because the recall is a little higher than the precision.

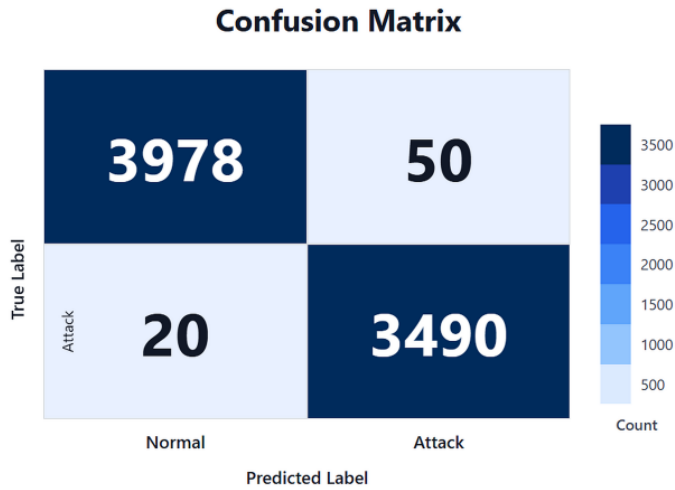

Figure 7. Confusion Matrix

4.2 Confusion Matrix

Figure 7 shows the confusion matrix of the proposed CNN model on the test set consisting of 7,538 samples. The model achieved 3,978 true negatives (TN), 50 false positives (FP), 20 false negatives (FN), and 3,490 true positives (TP). These results indicate that the model was able to distinguish normal and attack traffic with a low false alarm rate and a low miss rate.

4.3 ROC Curve

Figure 8 presents the ROC curve of the proposed CNN model. The curve characteristics indicate that the model can achieve high TPR (>0.95) at relatively low FPR (<0.05), demonstrating strong attack detection capabilities with minimal false positive rates.

4.4 Training and Validation Loss

The training history chart tracks how the model's loss (error) decreased over nine training epochs for both the training set (orange line) and validation set (blue line). Both curves start high—around 1.6 for training and 0.95 for validation—and drop steeply during the first few epochs as the model rapidly learns patterns in the data. By epoch 5, both losses have converged to very low values around 0.15 and continue to flatten out through the remaining epochs. The key insight here is that the training and validation losses move together without divergence, which indicates healthy learning without overfitting.

4.5 Performance Comparison CNN and Other Baselines

Table 3 shows a performance comparison among CNN and other baselines. CNN is outperforming other methods at accuracy, recall and f1. Whereas decision tree is the

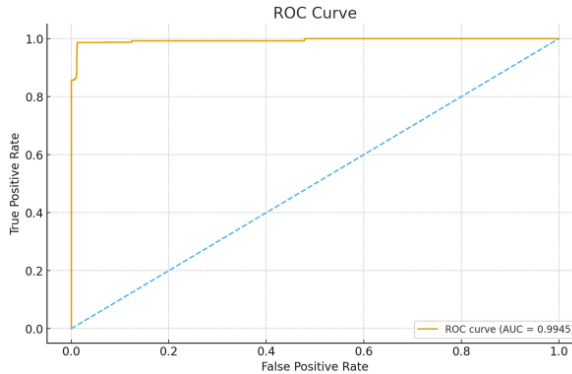

Figure 8. ROC Curve

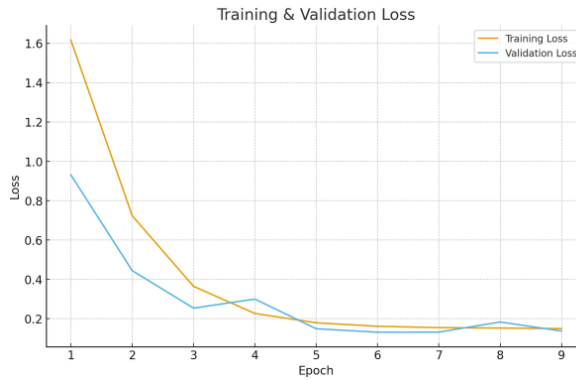

Figure 9. Training and Validation Loss

best at precision, XGBoost is the best at AUC. Autoencoder is the worst, with only 92.31% accuracy, which is a big drop from the others. This 6–7% difference may not seem like much, but it represents a lot more mistakes in forecasts in real life. It looks like the autoencoder's way of finding anomalies in this data doesn't work as well as the other techniques. This conclusion is incredibly useful since it shows us that not all advanced techniques will work for every problem.

Table 3. Performance Comparison CNN and Other Baselines

Method	Accuracy	Precision	Recall	F1	AUC
KNN	98.77%	98.76%	98.77%	98.69%	98.94%
Decision Tree	98.97%	98.97%	98.95%	98.90%	99.31%
XGBoost	98.65%	98.62%	98.67%	98.58%	99.77%
Autoencoder	92.31%	92.82%	92.66%	92.41%	92.48%
CNN	99.07%	98.59%	99.43%	99.00%	99.05%

4.6 Performance Comparison Between CNN and Recent State-of-the-Art Models

In this study, we also compared the proposed CNN model with two recent deep learning baselines, namely a Transformer based model and a TabNet inspired attention model. The results are presented in Table 4. As shown in the table, the proposed CNN achieved the best overall performance in terms of accuracy, recall, and F1- score. In particular, the CNN obtained an accuracy of 99.07%, a recall of 99.43%, and an F1-score of 99.00%, indicating a more balanced detection capability than the other models.

The Transformer model also showed strong performance, with an accuracy of 98.91% and the highest precision and AUC values, reaching 99.10% and 99.65%, respectively. Meanwhile, the TabNet inspired attention model produced lower results across most evaluation metrics, especially in recall and F1-score. Overall, these results indicate that the proposed CNN provides the most effective and balanced performance for the attack detection task in this study.

Table 4. Performance comparison between CNN and recent deep learning models

Method	Accuracy	Precision	Recall	F1-score	AUC
Transformer	98.91%	99.10%	98.60%	98.85%	99.65%
TabNet-inspired Attention	91.46%	95.85%	85.62%	90.45%	98.89%
CNN	99.07%	98.59%	99.43%	99.00%	99.05%

The Transformer based model and the TabNet inspired attention model were selected as recent deep learning baselines because both represent modern architectures for structured network traffic classification. All models were evaluated on the same dataset split, preprocessing pipeline, and evaluation metrics to ensure a fair comparison. This comparison was intended to position the proposed CNN not only against classical machine learning baselines, but also against more recent deep learning approaches.

5. Conclusion

This research introduces an anomaly based intrusion detection system designed to safeguard the GTP protocol in cellular networks through the application of convolutional neural networks. We created a specific dataset with both regular and harmful GTP traffic, and then we trained a CNN model to find different types of attacks, such as flood, invalid TEID, spoofing and malformed. The findings demonstrate that our CNN based method works very well, with an accuracy of 99.07%, a recall of 99.43%, and an AUC of 99.05%. This shows that it can find bad traffic while keeping false positives to a minimum.

The proposed CNN also outperformed the selected baseline methods, including KNN, Decision Tree, XGBoost, and Autoencoder, in terms of overall detection balance. The high recall rate is very significant for real world use because missing an actual attack could have serious effects on both network operators and their subscribers. It is hard for standard signature based systems to identify small changes that could mean new or changing attack methods. The model can learn spatial patterns from 18 extracted features, which helps it do this.

The outcomes are good, but this work has its limitations. Since the dataset was generated using UERANSIM and Open5GS simulators, it may not fully capture the complexity of real operator traffic. Live networks carry a much wider variety of traffic patterns, vendor implementations, and user behaviors that a simulated environment simply cannot replicate. Testing the model against real production traffic is therefore an important next step before any practical deployment. Future work could also explore combining this anomaly based approach with signature based detection, extend the model to classify specific attack types rather than just flagging anomalies, and improve its performance under high throughput network conditions. As 5G networks continue to grow, adaptive and robust security systems will be essential for protecting both operators and their subscribers.

Acknowledgement

Special thanks to Department of Electrical Engineering Universitas Indonesia for supporting this research and LPDP (Indonesia Endowment Fund for Education) for funding this research through their scholarship program.

Funding Statement This research was supported by grants from the LPDP (Indonesia Endowment Fund for Education) scholarship.

References

[1] S. Mavoungou et al. "Survey on Threats and Attacks on Mobile Networks". In: *IEEE Access* 4 (2016), pp. 4543–4572.

[2] M. Humayun et al. "5G Network Security Issues, Challenges, Opportunities and Future Directions: A Survey". In: *Journal of Physics: Conference Series* 1979 (2021), p. 012037.

[3] Y. et al. Zhang. "Invade the Walled Garden: Evaluating GTP Security in Cellular Networks". In: *Proceedings of the IEEE Symposium on Security and Privacy (SP)*. 2025.

[4] A. Thakkar and R. Lohiya. "A Review of the Advancement in Intrusion Detection Datasets". In: *Procedia Computer Science* 167 (2020), pp. 636–645.

[5] N. K. S. Nayak and R. Kumar. "An Intrusion Detection System for 5G SDN Network Using Binarized Deep Spiking Capsule Fire Hawk Neural Networks and Blockchain". In: *Future Internet* 16.10 (2024), p. 359.

[6] O. Malkawi, N. Obeid, and W. Almobaideen. "Intrusion Detection System for 5G Device-to-Device Communication Technology in Internet of Things". In: *Informatica* 48.15 (Oct. 2024), pp. 281–296.

[7] A. et al. Elhanashi. "Machine Learning Techniques for Anomaly-Based Detection System on CSE-CIC-IDS2018 Dataset". In: *Applications in Electronics Pervading Industry, Environment and Society (ApplePies 2022)*. Vol. 1036. Lecture Notes in Electrical Engineering (LNEE). Springer, 2023, pp. 131–140.

[8] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani. "Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization". In: *Proceedings of the 4th International Conference on Information Systems Security and Privacy (ICISSP)*. CIC-IDS2017 Dataset. 2018, pp. 108–116.

[9] R. I. Farhan, A. T. Maolood, and N. F. Hassan. "Performance Analysis of Flow-Based Attacks Detection on CSE-CIC-IDS2018 Dataset Using Deep Learning". In: *Indonesian Journal of Electrical Engineering and Computer Science* 20.3 (Dec. 2020), pp. 1413–1418.

[10] S. et al. Samarakoon. *5G-NIDD: A Comprehensive Network Intrusion Detection Dataset Generated over 5G Wireless Network*. arXiv preprint arXiv:2212.01298. Dec. 2022.

[11] P. et al. Radoglou-Grammatikis. "5GCIDS: An Intrusion Detection System for 5G Core with AI and Explainability Mechanisms". In: *Proceedings of the IEEE Globecom Workshops (GC Wkshps)*. Kuala Lumpur, Malaysia, 2023, pp. 353–358.

[12] Karsten Nohl and Luca Melette. *Advanced Interconnect Attacks – Chasing GRX and SS7 Vulnerabilities*. Chaos Communication Camp. Available at: https://media.ccc.de/v/camp2015-6785-advanced_interconnect_attacks. Mildenberg, Germany, Aug. 2015.

[13] M. Tanhatalab et al. *Deep RAN: A Scalable Data-Driven Platform to Detect Anomalies in Live Mobile Network Using Recurrent Convolutional Neural Network*. arXiv preprint arXiv:1911.04472. 2019.

[14] GSMA. *FS.20 – GPRS Tunnelling Protocol (GTP) Security*. Tech. rep. Version 4.0. https://www.gsma.com/fraud-security/resources/fs-20-gprs-tunnelling-protocol-gtp-security. London, United Kingdom: GSM Association, 2021.

[15] A. Almutairi and N. Abdelmajeed. "Innovative Signature Based Intrusion Detection System: Parallel Processing and Minimized Database". In: *2017 International Conference on the Frontiers and Advances in Data Science (FADS)*. 2017, pp. 114–119.

[16] Y. Yao et al. "Anomaly Intrusion Detection Approach Using Hybrid MLP/CNN Neural Network". In: *Sixth International Conference on Intelligent Systems Design and Applications*. 2006, pp. 1095–1102.

[17] B. Hussain et al. "Artificial Intelligence-Powered Mobile Edge Computing-Based Anomaly Detection in Mobile Networks". In: *IEEE Transactions on Industrial Informatics* 16.8 (2019), pp. 4986–4996.

[18] I. Saputra, E. Utami, and A. Muhammad. "Comparison of Anomaly Based and Signature Based Methods in Detection of Scanning Vulnerability". In: *2022 9th International Conference on Electrical Engineering, Computer Science and Informatics (EECSI)*. 2022, pp. 221–225.